Guide d’étapes clés – Débuggez le site d'une agence d'évènementiel

Comment utiliser ce document ?

Sur cette page, vous trouverez un exemple de découpage en étapes pour faire votre projet. Vous y trouverez :

- des recommandations pour réussir chaque étape ;
- les problèmes récurrents et points de vigilance à garder en tête ;
- une estimation de votre avancement sur l’ensemble du projet.

Ce découpage est simplement une suggestion pour vous organiser. Vous n’êtes pas obligé de compléter les étapes dans l’ordre.

Gardez en tête que votre progression sur les étapes n’est qu’une estimation, et sera différente selon votre vitesse de progression.

Étape 1 : Mettez en place votre environnement de développement

15 % de progression

Avant de démarrer cette étape, je dois avoir :

- Lu le scénario en entier au moins une fois.

Une fois cette étape terminée, je devrais pouvoir :

- Lancer l’application en mode développement via la commande **yarn start**.
- Lancer les tests via la commande **yarn test —watch**.

Recommandations :

- installez Node.js avec une version au moins égale à 16.14.2 LTS / npm 8.5.5 ;
- clonez le repository contenant le projet ;
- installez Yarn ;
- lancez la configuration des packages en suivant le guide inclus dans le README.md ;
- vérifiez que chaque commande ne renvoie pas d'erreur.

Ressources :

- Suivez le chapitre 1 de la partie 2 du cours *Débutez avec React* : Prenez en main votre app avec les commandes. Vous y trouverez tout le nécessaire pour installer correctement votre environnement de développement React, et lancer le projet.

Étape 2 : Installez React Developer Tools

30 % de progression

Certains outils nous permettent de constater l'évolution des valeurs, des assignations et des interactions entre les variables et les fonctions, en exécutant le programme pas à pas. La plupart des navigateurs web proposent des outils pour les développeurs, dont le debugger. En utilisant ReactJS, vous disposez d'un outil supplémentaire pour le debug, **React Developer Tools**.

Avant de démarrer cette étape, je dois avoir :

- Installé et configuré React Dev Tools.
- Exécuté l'application via **yarn start**.

Une fois cette étape terminée, je devrais pouvoir :

- Parcourir l'arbre de l'application React avec le tool Components. Cela va permettre de consulter en temps réel la propagation du state/context dans l'application et ses composants.

Recommandations :

- Manipulez l'outil et parcourez l'arbre pour comprendre le fonctionnement de l'application.
- Arrêtez-vous sur le composant **Slider** pour observer son comportement et son state. À chaque changement de slide, une valeur de l'état change. Notez bien sa valeur pour mieux comprendre le bug d'affichage du slide.

- Faites de même avec les autres composants du site qui posent problème.

Ressources :
- Installez React Developer Tools en suivant ce lien. Cet outil vous permettra de contrôler l'état de votre application en temps réel.

Étape 3 : Réparez les bugs

80 % de progression

Avant de démarrer cette étape, je dois avoir :
- Listé l'ensemble des bugs et localisé leur provenance dans le code.
- Lancé **yarn test —watch** pour consulter l'ensemble des tests restant en échec.

Une fois cette étape terminée, je devrais avoir :
- Validé tous les tests (0 test en failed) ;
- Réparé tous les bugs ;
- Un site qui répond aux besoins du client.

Recommandations :
- Lors de la réparation des bugs, vérifiez que le comportement de chaque composant est conforme aux attentes.

Ressources :
- Le cours Déboguez l'interface de votre site internet vous donnera des méthodes à mettre en pratique pour réparer les bugs.
- Le chapitre Partagez vos données avec le Contexte et useContext du cours *Créez une application React complète* vous permettra de modifier les données exposées par le Contexte DataContext.

Étape 4 : Rédigez le cahier de recette

100 % de progression

Avant de démarrer cette étape, je dois avoir :
- Réalisé l'ensemble des corrections.

Une fois cette étape terminée, je devrais avoir :

- Une liste de scénarios de tests à réaliser afin de confirmer que toutes les fonctionnalités attendues sont opérationnelles.

Recommandations

1. Reprenez une à une les fonctionnalités qui posent problèmes décrites par Jean-Baptiste.
2. Complétez le cahier de recette que l'ancien développeur a commencé à remplir, en suivant le même format.
3. Le nombre de scénarios n'est pas limité. Ils doivent permettre de répondre au besoin du client.
4. .

Ressources :

- Suivez le chapitre Apprenez le behavior driven development (BDD) du cours *Testez l'interface de votre site*. Ce cours vous aidera à formaliser la suite de tests dans votre cahier de recette.

: Ajoutez des tests additionnels

Pour aller plus loin, vous pouvez **écrire des tests unitaires** (pour vérifier le bon fonctionnement d'une partie ciblée du code) **et des tests d'intégration** (pour vous assurer que les différents composants du site fonctionnent bien entre eux). Ce sont des pratiques courantes, qui permettent d'améliorer la qualité du code et d'éviter les régressions (baisse du niveau de fonctionnalité). Faites cet exercice pour enrichir votre portfolio !

Avant de démarrer cette étape, je dois avoir :

- Repéré les parties de test manquantes.

Une fois cette étape terminée, je devrais avoir :

- 3 tests d'intégration de la page Home.
- 3 tests unitaires sur des composants ou helpers.

Recommandations :

- Réalisez les tests un à un.

Ressources :

- Consultez la documentation de Jest pour installer l'environnement de Jest et

réaliser les tests.

- Suivez le chapitre 1 de la partie 3 du cours *Testez vos applications Front End avec JavaScript*. Ce cours vous permettra d'écrire les tests unitaires attendus.
- Suivez le chapitre 2 de la partie 4 du cours *Testez vos applications Front End avec JavaScript*. Ce cours vous aidera à compléter les tests d'intégration attendus.

Projet terminé !